[10191/2275]

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: : Examiner: David Garcia Cervetti

Martin Hurich

For: METHOD AND DEVICE FOR DATA ENCRYPTION IN PROGRAMMING OF CONTROL UNITS

Filed: March 4, 2002 : Art Unit: 2136

Serial No.: 10/090,718

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to:
Mail Stop APPEAL BRIEF - PATENTS
Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450
on
Date: 2/24/2009
Signature:
AARON C. DEDITCH
(33,865)

APPEAL BRIEF PURSUANT TO 37 C.F.R. § 41.37

SIR:

In the above-identified patent application ("the present application"), Appellants electronically filed a Notice Of Appeal on October 24, 2008 from the Final Office Action issued by the U.S. Patent and Trademark Office on June 2, 2008, so that the two-month appeal brief due date is December 24, 2008, which has been extended by two months to February 24, 2009 by the accompanying Appeal Brief Transmittal and Petition to Extend.

In the Final Office Action, pending and considered *claims 1 to 12 and 15 to19 were finally rejected*. A Response After A Final Office Action (no amendments were made) was mailed on September 9, 2008, and an Advisory Action was mailed on September 25, 2008.

It is understood for purposes of the appeal that any Amendments to date have already been entered by the Examiner, and that the Response After Final does not require entry since it included no amendments.

As to the length of the "concise explanation" of the subject matter defined in each of the claims involved in the appeal (see 41.37), the "concise explanation" language is like the "concise explanation" requirement of former Rule 37 CFR 1.192. Accordingly, the length of the concise explanation provided is therefore acceptable, since it would have been acceptable under 37 CFR 1.192 and since it specifically defines the subject matter of the independent claims involved in the appeal. In the filing of many appeal briefs for the present Assignee , the length of the final "concise explanation" has always been finally accepted by the Patent Office.

It is therefore respectfully submitted that this Appeal Brief complies with 37 § C.F.R. 41.37. Although no longer required by the rules, this Brief is submitted in triplicate as a courtesy to the Appeals Board.

It is respectfully submitted that the final rejections of *claims 1 to 12 and 15 to19* should be reversed for the reasons provided below.

1. REAL PARTY IN INTEREST

The real party in interest in the present appeal is Robert Bosch GmbH ("Robert Bosch") of Stuttgart in the Federal Republic of Germany. Robert Bosch is the assignee of the entire right, title and interest in the present application.

2. RELATED APPEALS AND INTERFERENCES

There are no interferences or other appeals related to the present application, which "will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal".

3. STATUS OF CLAIMS

CLAIMS 13 and 14 ARE CANCELED.

A. Claims 1 to 12 and 15 to 19 were rejected under 25 U.S.C. 102(b) as anticipated by the U.S. Patent No. 6,141,421 ("Takaragi") reference.

Appellants therefore appeal from the final rejections of pending *claims 1 to 12 and 15 to 19*. A copy of all of the pending and appealed *claims 1 to 12 and 15 to 19* is attached hereto in the Claims Appendix.

4. STATUS OF AMENDMENTS

In response to the Final Office Action mailed on June 2, 2008, Appellants filed a Response After A Final Office Action (with no amendments), which was mailed on September 9, 2008.

It is understood for purposes of the appeal that any Amendments to date have already been entered by the Examiner, and that the Response After Final does not require entry since it included no amendments.

5. SUMMARY OF CLAIMED SUBJECT MATTER

The concise explanation of the summary of the claimed subject matter is as follows, as described in the context of the present application.

As to the presently claimed subject matter of claims 1, 7, 15 and 16, the specification describes and discloses the following:

Figure 1 shows a device having a programming unit 10, a control unit 11 and a data line 12. The programming unit 10 has a microprocessor 13 and a memory element 14 linked together by data bus 15. Control unit 11 has a microprocessor 16, a memory module 17 and a data bus 18. Data for programming control unit 11 is stored in memory module 14 of programming unit 10. The data is encrypted by microprocessor 13 by using a table and a hash function stored in memory module 14. This method uses the reversible operations of: rotation to the left (within a byte); rotation to the right (within a byte); and byte-by-byte XOR. (See Specification, page 5, lines 1 to 22).

The results here are invariant as to rotation by multiples of 8. For encryption, a table S having m elements S_0 through S_{m-1} is used. This table is accessed by a hash function h(x), which is an index. For a simpler description, the successive bytes during encryption are provided with an index i, where i = 0, 1, 2, ... An encrypted byte n* is formed from an unencrypted byte n according to the following (a starting value n_{-1} is used for decryption and encryption):

$$n_{-1} \equiv S_o \qquad \text{(formula 1)}$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^*\right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)} \qquad \text{(formula 2)}$$

The encrypted data is then transmitted via data line 12 to control unit 11. If the data is intercepted during transmission, it is harmless because the encrypted data cannot be used without the key, which is not transmitted along with the data. (See Specification, page 5, line 24 to page 6, line 9).

The encrypted data is stored in memory module 17 which contains the same key as that in memory module 14 of programming unit 10. The data is decrypted again with this key. Unencrypted byte n is formed from an encrypted byte n* according to:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)} \right) >>> \sum_{j=0}^{i} n_{j-1}^* \qquad \text{(formula 3)} .$$

Then control unit 11 may be programmed. The decrypted data is processed by microprocessor 16. (See Specification, page 6, lines 11 to 22).

The flow chart in Figure 2 shows the sequence of the method according to the presently claimed subject matter. In step 20, the encryption of the data, which is for programming the control unit, is performed first. *The data to be encrypted is not broken down into first and second words -- as in the related art*. Therefore, this method may also be used for individual bytes. This method uses a rotation about a number of positions which depends on the entire history of the encryption. The encryption of a byte is not predetermined but instead depends on the history. The elements of the field are not used in linear sequence but instead are selected by a hash function. The linking is not additive, but instead is accomplished by an XOR operation. An additional parameter is not the number of the operation but instead the selection of the hash function. This greatly reduces operating time. (See Specification, page 6, line 24 to page 7, line 11).

Data transmission via data line 12 occurs in a subsequent step 21. Since the transmitted data is encrypted, it is of no use for a possible hacker. Then in step 22 the data is entered/stored in memory module 17 of control unit 11. Then in step 23, the data is decrypted. The key for decryption is stored as a computer program in the memory module of control unit 11. The same key is used for decryption as the key used for encryption. This is a symmetrical method. (See Specification, page 7, lines 13 to 25).

Compared to the method in which wandering XOR masks are used for encryption, the key is not transmitted in the data stream or together with other parameters but instead is already present in the control unit. Also, no table is generated from pseudo-random numbers based on such parameters. In contrast with other methods, there is no addition of key values during the encryption and decryption. *In the method according to the presently claimed* *subject matter, the input values are not distributed among two or more registers, so they can be altered simultaneously thereafter*. (See Specification, page 7, line 27 to page 8, line 1).

The method according to the presently claimed subject matter is characterized in that the key, namely in this case the table and the hash function, are not transmitted over the data line in the data stream but instead are already present in the control unit. The method described here does not distribute the input values to two or more registers so they can be altered simultaneously thereafter, so it may also be used for individual bytes. This is particularly advantageous within flash programming. (See Specification, page 8, lines 3 to 11).

The method according to the presently claimed subject matter is typically used by a plurality of users. Therefore, that inadvertent data exchange between different users is to be prevented. This is avoidable because the method described here is parameterizable. With this method it is possible to safely encrypt large domains having the same content (filling areas). The encrypted domains do not provide any information regarding the key used. A byte-wise allocation between input and output data is impossible. (See Specification, page 8, lines 14 to 21).

This method does not require a temporary memory location for the data stream or parts thereof. Only one byte is needed as the memory location for the running total. The code demand for decryption is very low (approx. 130 bytes). This is extremely important for use in automotive control units. This method may also be adapted to different requirements by using a user-specific table. In addition, any desired hash function may be used for table access to make possible inferences regarding the content of the table difficult. The data throughput with the system described here can be as high as 7 MB/minute, for example. (See Specification, page 8, lines 22 to 36).

The presently claimed subject matter of claim 1 (and also essentially claims 11, 15 and 16) involves data encryption in programming of a control unit including: encrypting a complete stream of data to be transmitted in a programming unit using a first key, wherein a byte by byte encryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs; transmitting the data that had been encrypted to the control unit via a data line; and decrypting the data that had been encrypted in the programming unit using a second key provided in the control unit; wherein: successive bytes during encryption are provided with an index i, where i = 0, 1, 2,..., an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^*\right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)}\right) >>> \sum_{j=0}^{i} n_{j-1}^*.$$

(See claim 1).

The presently claimed subject matter of claim 7 includes essentially the same features as claim 1, except that it is to a data encryption system. (See claim 7).

The presently claimed subject matter of claim 11 includes essentially the same features as claim 1, except that it is to a computer program product. (See claim 11).

The presently claimed subject matter of claim 15 includes essentially the same features as claim 1, except that it is to a computer readable medium. (See claim 15).

The presently claimed subject matter of claim 16 includes essentially the same features as claim 1, except that it is to a computer readable medium, and it is directed to decryption. (See claim 16).

Finally, the appealed claims include no means-plus-function or step-plus-function claims, so that 41.37(v) is satisfied as to its specific requirements for such claims, since none are present here. The present application does not contain any step-plus-function claims because the method claims in the present application are not "step plus function" claims because they do not recite "a step for", as required by the Federal Circuit and as stated in Section 2181 of the MPEP.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether claims 1 to 12 and 15 to 19 are patentable under 25 U.S.C. 102(b) and therefore not anticipated by the U.S. Patent No. 6,141,421 ("Takaragi") reference.

7. ARGUMENT

A. The Rejections Under 35 U.S.C. § 102(b) That Claims 1 to 12 and 15 to 19 Are Anticipated

CLAIMS 1 TO 12 AND 15 TO 19

Claims 1 to 12 and 15 to 19 were rejected under 25 U.S.C. 102(b) as anticipated by the U.S. Patent No. 6,141,421 ("Takaragi") reference. .

As regards the anticipation rejections of the claims, to reject a claim under 35 U.S.C. § 102, the Office must demonstrate that each and every claim feature is identically described or contained in a single prior art reference. (See *Scripps Clinic & Research Foundation v. Genentech, Inc.*, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991)). As explained herein, it is respectfully submitted that the Office Actions to date do not meet this standard, for example, as to all of the features of the claims. Still further, not only must each of the claim features be identically described, an anticipatory reference must also enable a person having ordinary skill in the art to practice the claimed subject matter. (See *Akzo, N.V. v. U.S.I.T.C.*, 1 U.S.P.Q.2d 1241, 1245 (Fed. Cir. 1986)).

As further regards the anticipation rejections, to the extent that the Office Action may be relying on the inherency doctrine, it is respectfully submitted that to rely on inherency, the Office must provide a "basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristics *necessarily* flows from the teachings of the applied art." (See M.P.E.P. § 2112; emphasis in original; and see *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Int'f. 1990)). Thus, the M.P.E.P. and the case law make clear that simply because a certain result or characteristic may occur in the prior art does not establish the inherency of that result or characteristic.

The Takaragi reference refers to a speed optimized method for ascertaining a highly secure hash value. A hash function that is used to ascertain the hash value includes the cyclical shifting of data to be encrypted. This hash function is based on the knowledge that modern microprocessors are able to complete shifts of data in one computing cycle. To this end, data are subdivided into data blocks. The length of the data blocks is oriented toward the size of the register of the microprocessor. For example, for a 32-bit microprocessor, data blocks having a length of 32 bits are selected. A fast computer, such as a 100 MHz computer may thus process 100 million data blocks in one second. Thus, very long, and therefore very secure hash values may be ascertained in a short computing time.

In contrast to this, according to the subject matter of claim 1 (and essentially of claims 7, 11, 15 and 16), an entire data stream is intermittently encrypted and decrypted. The encrypting ensues using a first key, and the decrypting ensues using a second key. The functions used for encrypting and decrypting use a cyclical shifting.

Thus, in contrast to Takaragi, according to the subject matter of claim 1 (and essentially of independent claims 7, 11, 15 and 16), each arriving encrypted byte is immediately decrypted and may be used immediately -- independently of the other transmitted encrypted bytes. In this case, the objective is to provide a simple encrypting method, which may also be used for microcontrollers having a low computing power and a small program memory. This is especially useful for the use of the method in control units such as those used in the automotive sector. Furthermore, the individual coding and decoding of individual bytes is very advantageous, in particular for the flash programming of these control units in the automotive sector.

The Takaragi reference does not describe an encrypting method, but rather a signature method, which uses a very secure, but also very complex (i.e., long) hash value. Since very many operations must be executed in succession in order to ascertain this hash value, the method of Takaragi is not appropriate for use in the flash programming of a control unit. Since a high computing power, for example, a 32-bit 100 MHz microprocessor, is additionally required for the ascertainment of the hash value according to Takaragi, the

Takaragi reference teaches away from the subject matter of claim 1 (and claims 11, 15 and 16).

For these reasons, claims 1, 7, 11, 15 and 16 are allowable over the Takaragi reference, as are their respective dependent claims.

Still further as to claim 1, it is respectfully submitted that the "Takaragi" reference does not identically disclose (or suggest) the feature in which:

> *successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,*
>
> *an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:*
>
> $$n_{-1} \equiv S_o$$
>
> $$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^*\right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)},$$
>
> *an unencrypted byte n is formed from an encrypted byte n* according to the following:*
>
> $$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)}\right) >>> \sum_{j=0}^{i} n_{j-1}^*.$$

The Final Office Action conclusorily asserts that "Takaragi" (col. 9, lines 30-45) discloses the feature of "cyclically shifting bits and exclusive OR operations". Even if "Takaragi" may refer to how a cyclical shift of bits and an XOR operation works, it does not identically disclose (or even suggest) the above cited equations for data encryption and decryption, as provided for in the context of the claimed subject matter. Other than referring to the use of two basic and fundamental binary operations, "Takaragi" (col. 9, lines 21-29) does not identically disclose (or even suggest) the above cited equations, as provided for in the context of the presently claimed subject matter.

The Office conclusorily asserts that "the equation is irrelevant, since "Takaragi" teaches what the equation tries to accomplish, encrypt and decrypt information." (Final Office Action, ¶ 5). The equation is not "irrelevant" but is a feature of the claim that must be "identically described or contained in a single prior art reference." The encryption/ decryption method of "Takaragi", which refers to XORs and bit-sifters, simply does not anticipate or render obvious the specific features of the presently claimed subject matter. The presently claimed subject matter reflects a new data encryption method <u>*using the encryption history*</u>, which is not identically disclosed (or even suggested) by the "Takaragi" reference.

The equations of the claimed subject matter accurately describe and claim part of a novel encryption method in which each encrypted unit relies on the encryption value of prior units.

The Final Office Action conveniently and conclusorily suggests that "with the richness of the English language, perhaps an explanation of what the equation does should replace said equation in the body of the claim." In fact, data encryption and decryption is a mathematically based operation and as "rich" as the English language may be, the mathematical formulas provided are the most accurate and appropriate way of "particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention."

The Final Office Action conclusorily asserts that "according to the specification, the operations of the equations are nothing more, nothing less, than rotation to the left and rotation to the right, and exclusive OR (see page 5)." (Final Office Action, ¶ 5). In fact, the specification discloses that those three operations are used by the specific method, and then (at the end of page 5 into page 6) describes the equations of the claimed subject matter.

The Final Office Action further states that "accordingly, "Takaragi"'s operations of 'cyclically shifting and x-OR operations' (see previous Office Action) correctly map to the claimed subject matter. In other words, the equations encrypt by rotating to the left and then xoring, and decrypt by xoring and then rotating to the right." (Final Office Action, ¶ 5). The present claims -- and not "Takaragi" -- present a new and nonobvious way to encrypt a stream of data based on prior bytes of the stream. The "Takaragi" encryption method, even if it may be using XORs and bit-shifting, does not identically disclose (or even suggest) these specific features, and therefore does not and cannot anticipate the present claims.

Also, Applicant simply does not admit or agree that "Takaragi" provides a basic architecture to implement the claimed subject matter. In particular, "Takaragi", does not identically disclose (nor even suggest) the specific implementation of data encryption of the present claims. Absolutely nothing in "Takaragi" even begins to anticipate (or render obvious) the novel encryption method of the claimed subject matter.

It is also respectfully submitted that the "Takaragi" reference does not identically disclose (or even suggest) the feature in which "no byte-wise allocation between input and output data occurs", as provided for in the context of the claimed subject matter. The Final Office Action asserts that "Takaragi" at col. 9, lines 43-47, discloses no byte-wise allocation because it states that "the 64-bit input data X_2 and the 32-bit input data Y_2 is expanded to data of total 128 bits." Thus even if "Takaragi" may concern byte-wise allocation, it does not identically disclose (or even suggest) the claim feature in which "no byte-wise allocation between input and output data occurs."

Additionally, it is respectfully submitted that the "Takaragi" reference does not identically disclose (or suggest) the feature in which "decrypting the data that had been encrypted in the programming unit using a second key provided in the control unit," as provided for in the context of the claimed subject matter. In Figure 20 (element 2017) it seems plain that the key is provided in the data stream, and in Figure 19 (element 1905) and Figure 18 (element 1802), it is plain that the key is external to element 1901 "Decode and Expansion Function" and element 1801 "Decoder". In "Takaragi", col. 14, line 59 states that the "a key 1905 is inputted", and is provided *to* the decoder. This does not identically disclose (or even suggest) the feature of "provided *in* the control unit". The Final Office Action claims that "fig. 19 is a function that resides in the decoder of fig. 18, and therefore the key is in the decoder. In fact, Fig. 19 is an expanded view of a function of Fig. 18 (see "Takaragi", brief description of the drawings). Element 1901 of Fig. 19 is in the decoder of element 1801 of Fig. 18, but elements 1902, 1920, and 1905 are plainly outside the decoder 1801 of Fig. 18. The key elements 1905 and 1802 are plainly outside the decoder, since they are provided to the decoder.

For at least these reasons, the reference does not identically disclose (or suggest) the above-discussed features of the claimed subject matter, so that claim 1 is allowable, as are its dependent claims 2 to 6 and 17.

Claim 7 includes features like those of claim 1 and is therefore allowable for essentially the same reasons, as are its dependent claims 8 to 10 and 18.

Claims 11 (and dependent claim 19), 15, and 16 include like those of claim 1, and are therefore allowable for essentially the same reasons.

It is therefore respectfully submitted that claims 1 to 12 and 15 to 19 are allowable.

CONCLUSION

In view of the above, it is respectfully requested that the rejections of finally rejected, pending and considered claims 1 to 12 and 15 to 19 be reversed since these claims are allowable.

Respectfully submitted,

Dated: 2/24/2009

By: G A Messina By AD Reg. no. 33,865 Aaron C. Deditch)
Gerard A. Messina
(Reg. No. 35,952)

KENYON & KENYON LLP
One Broadway
New York, New York 10004
(212) 425-7200

CUSTOMER NO. 26646

1689616

CLAIMS APPENDIX

1. (Previously Presented) A method of data encryption in programming of a control unit comprising:

encrypting a complete stream of data to be transmitted in a programming unit using a first key, wherein a byte by byte encryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs;

transmitting the data that had been encrypted to the control unit via a data line; and

decrypting the data that had been encrypted in the programming unit using a second key provided in the control unit;

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^*\right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)}\right) >>> \sum_{j=0}^{i} n_{j-1}^*.$$

2. (Original) The method of claim 1, wherein the first key and the second key are identical.

3. (Original) The method of claim 1, wherein the first key and the second key are not identical.

4. (Original) The method of claim 2, wherein each one of the first key and the second key includes a table that is accessed by a hash function.

5. (Original) The method of claim 1, wherein at least one of the first key and the second key is implemented in an electronic circuit.

6. (Original) The method of claim 1, wherein at least one of the first key and the second key is implemented in the form of a computer program.

7. (Previously Presented) A data encryption system, comprising:

a programming unit in which a first key is provided;

a control unit in which a second key is provided; and

a data line coupled to the programming unit and the control unit for transmitting encrypted data, the encrypted data being an encryption of a complete stream of data, wherein a byte by byte encryption of the complete stream of data is capable of being performed, wherein encryption of a byte includes a rotation of bits of the byte about a number of positions, the number depending on an entire history of the encryption of the data, and wherein no byte-wise allocation between input and output data occurs;

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^*\right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n^*_{j-1}\right)}\right) >>> \sum_{j=0}^{i} n^*_{j-1}$$
.

8. (Original) The system of claim 7, wherein the first key and the second key are identical.

9. (Original) The system of claim 7, wherein the first key and the second key are not identical.

10. (Original) The system of claim 7, wherein the programming unit and the control unit each includes an electronic computing unit and a memory module that are linked together by a data bus.

11. (Previously Presented) A computer program product having program code executable by a computing unit, the program code when executed causing the computing unit to perform a method, the method comprising:

performing an encryption of a complete stream of data in accordance with a table and a hash function, wherein a byte by byte encryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs:

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n^*_{j-1}\right) \oplus S_{h\left(\sum_{j=0}^{i} n^*_{j-1}\right)}$$
,

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)} \right) >>> \sum_{j=0}^{i} n_{j-1}^*$$

12. (Previously Presented) The computer program product of claim 11, wherein the computing unit includes an electronic computing unit in a programming unit.

13. (Canceled).

14. (Canceled).

15. (Previously Presented) A computer-readable medium, comprising:

a program code executable on a computing unit for performing an encryption of a complete stream of data in accordance with a table and a hash function, wherein a byte by byte encryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs, as provided for in the context of the claimed subject matter:

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o \quad .$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^* \right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)} ,$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n^*_{j-1}\right)} \right) >>> \sum_{j=0}^{i} n^*_{j-1}$$.

16. (Previously Presented) A computer-readable medium, comprising:

a program code executable on a computing unit for performing a decryption of a complete stream of data in accordance with a table and a hash function, wherein a byte by byte decryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs:

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n^*_{j-1} \right) \oplus S_{h\left(\sum_{j=0}^{i} n^*_{j-1}\right)}$$,

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n^*_{j-1}\right)} \right) >>> \sum_{j=0}^{i} n^*_{j-1}$$.

17. (Previously Presented) The method of claim 1, wherein there is no bit-wise allocation between input and output data:

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . ., an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^*\right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)}\right) >>> \sum_{j=0}^{i} n_{j-1}^*.$$

18. (Previously Presented) The method of claim 7, wherein there is no bit-wise allocation between input and output data.

19. (Previously Presented) The method of claim 11, wherein there is no bit-wise allocation between input and output data.

EVIDENCE APPENDIX

Appellants have not submitted any evidence pursuant to 37 CFR Sections 1.130, 1.131 or 1.132, and do not rely upon evidence entered by the Examiner.

RELATED PROCEEDINGS INDEX

There are no interferences or other appeals related to the present application.